A1

RENDERING DIGITAL IMAGES HAVING SURFACE REFLECTANCE PROPERTIES," the disclosures of which are hereby incorporated herein by reference.

In the claims:

Pursuant to 37 C.F.R. § 1.121(c)(1), applicant has set forth amendments to claim 1 by rewriting claim 1 with all changes. Also, applicant has included all pending claims, whether amended or unchanged, for the convenience of the Examiner.

Sub B8

1. A method for rendering a three-dimensional (3D) graphical image, said method comprising the steps of:

representing said 3D graphical image as a plurality of graphics primitives each having a plurality of vertices;

A2

for each of said plurality of graphics primitives, computing at least two texture coordinate gradient vectors;

for each vertex of said plurality of graphics primitives, determining a 3D coordinate frame, wherein said determining step includes using said at least two texture coordinate gradient vectors computed for the respective graphics primitive for orienting said 3D coordinate frame; and

utilizing at least said 3D coordinate frame to determine parameters of a parametric texture mapping function.

2. The method of claim 1 further comprising the step of:

evaluating said parametric texture mapping function for rendering said 3D graphical image.

3. The method of claim 2 wherein said parametric texture mapping function comprises a biquadric polynomial having at least six coefficients.

4. The method of claim 1 wherein said step of utilizing comprises:

calculating scalar components for said parametric texture mapping function.

5. The method of claim 4 wherein said scalar components include lighting scalar components.

6. The method of claim 1 wherein said 3D coordinate frame is formed by a normal vector, tangent vector, and binormal vector.

7. The method of claim 6 wherein said step of utilizing further comprises:

calculating a first lighting scalar component for said parametric texture mapping function as the dot product of a light vector and said tangent vector; and

calculating a second lighting scalar component for said parametric texture mapping function as the dot product of said light vector and said binormal vector.

B8 cont

8. The method of claim 1 wherein said step of computing at least two texture coordinate gradient vectors comprises:

computing a first texture coordinate gradient vector that identifies the direction of maximum change along a first texture coordinate; and

computing a second texture coordinate gradient vector that identifies the direction of maximum change along a second texture coordinate.

9. The method of claim 1 wherein said step of computing at least two texture coordinate gradient vectors comprises:

assigning a computed value to a variable; and

utilizing said variable in computing each of said at least two texture coordinate gradient vectors.

10. The method of claim 9 wherein said utilizing step further comprises:

using said variable as a denominator in calculating linear combination variables that are used in further computing said at least two texture coordinate gradient vectors.

11. The method of claim 1 wherein said parametric texture mapping function is a luminance parametric texture mapping function.

12. The method of claim 1 wherein said parametric texture mapping function is a red-green-blue (RGB) parametric texture mapping function.

13. The method of claim 1 wherein said graphics primitive comprises a polygon.

14. A method for mapping a computer graphics texture to a three-dimensional (3D) object, said method comprising the steps of:

representing said 3D object as a plurality of graphics primitives;

determining a first vector from a first vertex of a graphics primitive to a second vertex of said graphics primitive;

determining a second vector from said first vertex to a third vertex of said graphics primitive;

B8 cont

calculating a first dot product of said first vector by said first vector;

calculating a second dot product of said first vector and said second vector;

assigning one variable a value derived from at least said first dot product and said second dot product;

computing at least two texture coordinate gradient vectors utilizing at least said one variable, wherein said at least two texture coordinate gradient vectors are indicative of orientation of a texture mapped to said graphics primitive;

determining a 3D coordinate frame for each vertex of said graphics primitive, wherein said determining comprises using said at least two texture coordinate gradient vectors for orienting said 3D coordinate frame; and

utilizing at least said 3D coordinate frame in mapping said texture to said 3D object.

15. The method of claim 14 wherein said mapping said texture to said 3D object comprises utilizing a parametric texture map function.

16. A system for rendering three-dimensional (3D) graphical images, said system comprising:

data structure representing said 3D graphical image as a plurality of graphics primitives each having a plurality of vertices;

software code executable to compute at least two texture coordinate gradient vectors for each of said plurality of graphics primitives;

software code executable to determine a 3D coordinate frame for each vertex of said plurality of graphics primitives, wherein said software code uses said at least two texture coordinate gradient vectors computed for the respective graphics primitive for orienting said 3D coordinate frame; and

B8 cont

software code executable to utilize at least said 3D coordinate frame to determine parameters of a parametric texture mapping function.

17. The system of claim 16 wherein said software code executable to compute at least two texture coordinate gradient vectors further comprises:

software code executable to assign a computed value to a variable; and

software code executable to utilize said variable in computing each of aid at least two texture coordinate gradient vectors.

18. The system of claim 16 wherein said software code executable to compute at least two texture coordinate gradient vectors further comprises:

software code executable to use said variable as a denominator in calculating linear combination variables that are used in further computing said at least two texture coordinate gradient vectors.

19. The system of claim 16 wherein said 3D coordinate frame is formed by a normal vector, tangent vector, and binormal vector.

20. The system of claim 19 wherein said software code executable to determine a 3D coordinate frame further comprises:

Cnl BA

software code executable to calculate a first lighting scalar component for said parametric texture mapping function as the dot product of a light vector and said tangent vector; and

software code executable to calculate a second lighting scalar component for said parametric texture mapping function as the dot product of said light vector and said binormal vector.